**Project YT**

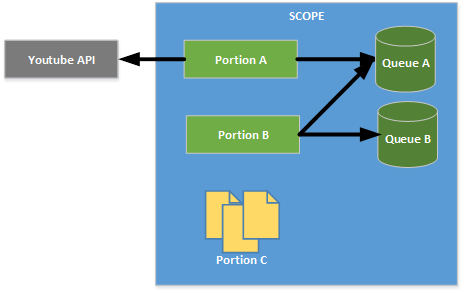
**Objectives** are to evaluate capabilities of developers concerning

a)       Show your coding skills

b)      Show  your capabilities to learn new technologies

c)       Show your capabilities to support a working product with required documentation to understand and execute the solution

**Scope:**



         Part A - build a solution in Java that will

o   Connect to youtube using the youtube APIs ([https://developers.google.com/youtube/](https://developers.google.com/youtube/" \t "_blank)) and retrieve all video metadata containing the word “telecom” in the title

o   For each video identified, publish an XML message in Queue A, containing at least the following information:

  URL

  Video title

         Part B- build a solution in Java that will

o   Consume the XML messages from JMS Queue A (output of Part A)

o   Modify the video Title in each XML message, replacing the word “telecom” with the word “telco”

o   Store the modified XML message in JMS queue B

         Part C - Provide a document describing

o   Which design patterns have you used in the implementation

o   Schema describing the messages produced by the solution

o   How to compile the source code produced

o   How to execute your programs to produce the results previously described

* Looking back at the implementation, what would you consider doing differently

**Which design patterns have you used in the implementation**

Factory (JMSFactory)

Singleton (JMSFactory)

Decoupling (JmsConsumer, JmsProducer, JMSFactory)

Bridge (JmsConsumerImpl, JmsProducerImpl)

Iterator (SearchYouTube, Producer)

**Schema describing the messages produced by the solution**

From YouTube to the producer (extract):

=============================================================

2019/10/27 08:16:26.725 DEBUG SearchYouTube - First 5 videos for search on "telecom".

2019/10/27 08:16:26.725 DEBUG SearchYouTube - =============================================================

2019/10/27 08:16:26.725 DEBUG SearchYouTube - Video Id: jUXudBMOZVE

2019/10/27 08:16:26.725 DEBUG SearchYouTube - Title : SKT vs SPY Highlights Game 1 | Worlds 2019 Quarter-finals | SK Telecom T1 vs Splyce G1

2019/10/27 08:16:26.725 DEBUG SearchYouTube - Thumbnail : https://i.ytimg.com/vi/jUXudBMOZVE/default.jpg

2019/10/27 08:16:26.725 DEBUG SearchYouTube -

-------------------------------------------------------------

From the producer to YouTube Queue A (extract):

<SearchResult>

<id>

<kind>youtube#video</kind>

<videoId>jUXudBMOZVE</videoId>

<URL>https://www.youtube.com/watch?v=jUXudBMOZVE</URL>

</id>

<snippet>

<thumbnails>

<default>

<url>https://i.ytimg.com/vi/jUXudBMOZVE/default.jpg</url>

</default>

</thumbnails>

<title>SKT vs SPY Highlights Game 1 | Worlds 2019 Quarter-finals | SK Telecom T1 vs Splyce G1</title>

</snippet>

</SearchResult>

From YouTube Queue A to YouTube Queue B (extract):

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<SearchResult>

<id>

<kind>youtube#video</kind>

<videoId>jUXudBMOZVE</videoId>

<URL>https://www.youtube.com/watch?v=jUXudBMOZVE</URL>

</id>

<snippet>

<thumbnails>

<default>

<url>https://i.ytimg.com/vi/jUXudBMOZVE/default.jpg</url>

</default>

</thumbnails>

<title>SKT vs SPY Highlights Game 1 | Worlds 2019 Quarter-finals | SK telco T1 vs Splyce G1</title>

</snippet>

</SearchResult>

**How to compile the source code produced**

You’ll need java 8 and Apache Maven installed.

Unzip ProjectYT.zip

Open a DOS console and type:

cd ProjectYT

mvn clean install

**How to execute your programs to produce the results previously described**

Open a DOS console and type:

cd ProjectYT

start\_broker.bat (if the port is already busy you can change it in the file /config.json under BROKER\_TCP\_ADDRESS, requires restart)

Once the JMS broker has started open another DOS console and type:

cd ProjectYT

start\_consumer.bat

Once the JMS consumer booted, open another DOS console and type:

cd ProjectYT

start\_producer.bat

The logs are in the console and in the ProjectYT / ProjectYT.log file.

The logs level goes from info to debug to better see the I / O (changeable in the file src\main\resources\log4j2.xml, requires recompilation and restart)

**Looking back at the implementation, what would you consider doing differently**

* The Config class by maybe could be a singleton
* A better use of the session object